This tutorial shows you how to train an Automated Speech Recognition (https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&as_vis=1&q=automatic+speech+recognition+technology&oq=%22automatic+speech+recognition%22+) (ASR) model using the publicly available Librispeech ASR corpus (http://www.openslr.org/12/) dataset with Tensor2Tensor (https://github.com/tensorflow/tensor2tensor) on a Cloud TPU.

The speech recognition model is just one of the models in the Tensor2Tensor (https://github.com/tensorflow/tensor2tensor) library. Tensor2Tensor (**T2T**) is a library of deep learning models and datasets as well as a set of scripts that allow you to train the models and to download and prepare the data. This model does speech-to-text conversion.

Before starting this tutorial, follow the steps below to check that your Google Cloud project is correctly set up.

1. Sign in (https://accounts.google.com/Login) to your Google Account.

   If you don't already have one, sign up for a new account (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

   ★ **Note**: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

   Go to the project selector page (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. Learn how to confirm billing is enabled for your project (/billing/docs/how-to/modify-project).

This walkthrough uses billable components of Google Cloud. Check the Cloud TPU pricing page (/tpu/docs/pricing) to estimate your costs. Be sure to clean up (#clean_up) resources you create when you've finished with them to avoid unnecessary charges.

This section provides information on setting up Cloud Storage storage, VM, and Cloud TPU resources for tutorials.

tant: Set up all resources in the same region/zone to reduce network latency and network costs.

You need a Cloud Storage bucket to store the data you use to train your model and the training results. The `ctpu up` tool used in this tutorial sets up default permissions for the Cloud TPU service account. If you want finer-grain permissions, review the access level permissions (/tpu/docs/storage-buckets).

The bucket location must be in the same region as your virtual machine (VM) and your TPU node. VMs and TPU nodes are located in specific zones (/tpu/docs/types-zones#types), which are subdivisions within a region.

1. Go to the Cloud Storage page on the Cloud Console.

   Go to the Cloud Storage page (https://console.cloud.google.com/storage/browser)

2. Create a new bucket, specifying the following options:

   - A unique name of your choosing.
   - Select `Region` for Location type and `us-central1` for the Location (zone)
   - Default storage class: `Standard`
   - Location: Specify a bucket location in the same region where you plan to create your TPU node. See TPU types and zones (/tpu/docs/types-zones#types) to learn where various TPU types are available.

This section demonstrates using the Cloud TPU provisioning tool (https://github.com/tensorflow/tpu/tree/master/tools/ctpu) (`ctpu`) for creating and managing Cloud TPU project resources. The resources are comprised of a virtual machine (VM) and a Cloud TPU resource that have the same name. **These resources must reside in the same region/zone as the bucket you just created.**

You can also set up your VM and TPU resources using `gcloud` commands or through the Cloud Console (https://console.cloud.google.com/). See the creating and deleting TPUs (/tpu/docs/creating-deleting-tpus) page to learn all the ways you can set up and manage your Compute Engine VM and Cloud TPU resources.

1. Open a Cloud Shell window.

   Open Cloud Shell (https://console.cloud.google.com/?cloudshell=true)

2. Run `gcloud config set project <var>your-project</var>` to set the project where you want to create Cloud TPU.

3. Run `ctpu up` specifying the flags shown for either a Cloud TPU device or Pod slice. If you do not specify `tpu-size`, the default is a v2-8 Cloud TPU. Refer to CTPU Reference (/tpu/docs/ctpu-reference) for flag options and descriptions.

4. Set up a Cloud TPU device:

★ **Note:** If you have more than one project, you must specify the project name. If `--name` is not specified, it defaults to your username. If `--zone` is not specified, it defaults to `us-central1-b`. Make sure the zone matches the zone where you set up the storage bucket.

5. The configuration you specified appears. Enter y to approve or n to cancel.

6. When the `ctpu up` command has finished executing, verify that your shell prompt has changed from `username@project` to `username@tpuname`. This change shows that you are now logged into your Compute Engine VM.

Note: If you are not connected to the Compute Engine instance, you can connect by running the following commands, replacing ***vm-name*** with the name of your VM:

As you continue these instructions, run each command that begins with `(vm)$` in your VM session window.

T2T conveniently packages data generation for many common open-source datasets in its `t2t-datagen` script. The script downloads the data, preprocess it, and makes it ready for training. To do so, it needs local disk space.

You can skip this step if you used `ctpu up` to create your Compute Engine VM since it provides 250 GB of disk space for your VM. If you set up your Compute Engine VM using `gcloud` commands or the Cloud Console, and did not specify the VM disk size to be at least 200 GB, follow the instructions below.

- Follow the Compute Engine guide to add a disk (/compute/docs/disks/add-persistent-disk) to your Compute Engine VM.
- Set the disk size to 200 GB (the recommended minimum size).
- Set **When deleting instance** to **Delete disk** to ensure that the disk is removed when you remove the VM.

Make a note of the path to your new disk. For example: `/mnt/disks/mnt-dir`.

It can take approximately 17 hours to generate the training and evaluation datasets.

On your Compute Engine VM:

1. Create the following environment variables for directories:

where:

- `YOUR-BUCKET-NAME` is the name of your Cloud Storage bucket.
- `DATA_DIR` is a location on Cloud Storage that holds the training and evaluation data.
- `OUT_DIR` specifies the directory where checkpoints and summaries are stored during model training. If the folder is missing, the program creates one. When using a Cloud TPU, the `output_dir` must be a Cloud Storage path (`gs://...`). You can reuse an existing folder to load current checkpoint data and to store additional checkpoints.
- `YOUR-TMP_DIRECTORY` is a location to use to store temporary data. If you added a disk to your Compute Engine VM, this will be a location on the added disk (for example, `/mnt/disks/mnt-dir/t2t_tmp`. Otherwise, it will be a temporary directory on your VM (for example, `/tmp/t2t_tmp`).

2. If you added a new disk to your Compute Engine VM, create a temporary directory on the added disk.

Note: You do not need to create a temporary directory if you are using a local directory on your Compute Engine VM.

3. Use the `t2t-datagen` (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/bin/t2t-datagen) script to generate both the full dataset and the small clean version, which you will use for evaluation.

   The audio import in `t2t-datagen` uses `sox` to generate normalized waveforms. Install it on your Compute Engine VM and then run the `t2t-datagen` commands that follow.

The problem `librispeech_train_full_test_clean` trains on the full dataset but evaluate on the clean dataset.

You can also use `librispeech_clean_small` which is a small version of the clean dataset.

You can view the data on Cloud Storage by going to the Google Cloud Console and choosing **Storage** from the left-hand menu. Click the name of the bucket that you created for this tutorial.

To train a model on Cloud TPU run the trainer with big batches and truncated sequences.

After this step is completed, run the training again for more steps with smaller batch size and full sequences. This training take approximately 11 hours on a `v3-8` TPU node.

To avoid incurring charges to your GCP account for the resources used in this topic:

1. Disconnect from the Compute Engine VM:

   Your prompt should now be `user@projectname`, showing you are in the Cloud Shell.

2. In your Cloud Shell, run `ctpu delete` with the –zone flag you used when you set up the Cloud TPU to delete your Compute Engine VM and your Cloud TPU:

★ **Important:** If you set the TPU resources name when you ran **`ctpu up`**, you must specify that name with the **`--name`** flag when you run **`ctpu delete`** in order to shut down your TPU resources.

3. Run `ctpu status` to make sure you have no instances allocated to avoid unnecessary charges for TPU usage. The deletion might take several minutes. A response like the one below indicates there are no more allocated instances:

4. Run `gsutil` as shown, replacing ***bucket-name*** with the name of the Cloud Storage bucket you created for this tutorial:

For free storage limits and other pricing information, see the Cloud Storage pricing guide (/storage/pricing).

- Learn more about `ctpu` (https://github.com/tensorflow/tpu/tree/master/tools/ctpu), including how to install it on a local machine.
- Explore more Tensor2Tensor models for TPU (https://github.com/tensorflow/tensor2tensor/blob/master/docs/cloud_tpu.md).
- Experiment with more TPU samples (/tpu/docs/tutorials/).
- Explore the TPU tools in TensorBoard (/tpu/docs/cloud-tpu-tools).